

In the Claims:

Please cancel claims 13 and 14 without prejudice.

Please amend the claims as indicated below.

1. (currently amended) A method for monitoring system processor resources utilized~~usage time~~ ~~of by~~ a software agent operating in a computer system, wherein said agent comprises an executable sequence of instructions, said method comprising the steps of:

identifying said agent, wherein said identifying is performed by a CPU resource tracking process determining that said agent is running;

~~by associating, by said CPU resource tracking process responsive to said identifying of said agent, an agent identifier with said agent therewith,~~ wherein said agent is one of a plurality of software agents operating in said computer system, and wherein said agent identifier uniquely identifies said agent within said plurality of software agents operating in said computer system;

initiating, responsive to said identifying of said agent, an agent lifetime timer for measuring an operating interval of said agent during which said agent is running in said computer system;

determining said operating interval using said agent lifetime timer by identifying a start time at which said CPU resource tracking process determined that said agent is running and a completion time ~~of said agent~~ at which said CPU resource tracking process determines that said agent has expired, and computing said operating interval~~an elapsed time~~ as the difference between said starting time and said completion time ~~for said agent~~; and

calculating an amount of said system processor resources utilized by said agent during said operating interval at least in part by detecting ~~creation of~~ a plurality of threads created by said agent during said operating interval, wherein said calculating said amount of said system processor resources utilized by said agent during said operational interval further includes

~~calculating determining~~ CPU usage for each of said plurality of threads and adding said ~~calculateddetermined~~ CPU usage for ~~alleach~~ of said plurality of threads together to ~~determine at~~ least a ~~portion of~~ said amount of system processor resources utilized by said agent during said operating interval, ~~wherein said calculating said CPU usage for each of said plurality of threads is~~ performed responsive to ~~as each~~ respective one of said plurality of threads ~~expiring, and~~ wherein said adding said calculated CPU usage for all of said plurality of threads together is performed by adding said calculated CPU utilization for each respective one of said plurality of threads responsive to each respective one of said plurality of threads ~~expiring~~, wherein each of said plurality of threads is a path of execution such that multiple of said plurality of threads can be executed simultaneously; and

storing said operating interval, said amount of system processor resources utilized by said agent during said operating interval and said agent identifier in a computer-readable memory.

2. (original) The method of claim 1, wherein said computer-readable memory includes a hash table.
3. (canceled)
4. (canceled)
5. (currently amended) The method of claim 1 further comprising:  
associating said operating interval and said agent identifier with other operating intervals and agent identifiers associated with other ones of said plurality of ~~with a plurality of other~~ software agents operating in said computer system.
6. (currently amended) The method of claim 5 further comprising:  
filtering said ~~agent and said plurality of~~ software ~~other~~ agents according to predefined filtering criteria to produce a filtered set.

7. (original) The method of claim 6 further comprising:  
rank ordering said filtered set.
8. (original) The method of claim 7 further comprising:  
making said filtered set available to a display device.
9. (currently amended) The method of claim 6 further comprising:  
determining a corrective measure for at least one member of said filtered set  
having an amount of system processor resources utilized that exceeds a predetermined threshold.
10. (original) The method of claim 9 further comprising:  
displaying said corrective measure on a display device.
11. (original) The method of claim 9, wherein said corrective measure is implemented by said system.
12. (canceled)
13. (cancelled)
14. (cancelled)
15. (currently amended) The method of claim ~~14~~3, further comprising:  
running a collection probe to determine if a total amount of consumed system processor resources exceeds a predetermined threshold~~a total amount of consumed system processor resources exceeds said configuration threshold~~; and  
performing said initiating step when said total amount of consumed system processor resources exceeds said configuration threshold.

16. (canceled)
17. (canceled)
18. (canceled)
19. (canceled)
20. (currently amended) The method of claim 1, wherein said agent comprises a web based event driven agent, and wherein said agent identifier comprises a name and an HTTP (HyperText Transport Protocol) thread identifier

~~A method for tracking system processor time of a target agent operatively associated with a hypertext transport protocol process operating on a computer system and running a plurality of threads, wherein said target agent comprises an executable sequence of instructions, said method comprising:~~

- ~~— creating a computer-readable hash table in a memory operatively associated with said computer system;~~
- ~~— initiating an agent tracking function in machine-executable code in said computer system;~~
- ~~— identifying a plurality of threads created by said target agent and associating a thread identifier with each of said plurality of threads to produce a~~  
~~an identified thread set;~~
- ~~— determining an amount of said system processor time utilized by said identified thread set by determining CPU usage for each of said plurality of threads and adding said determined CPU usage for each of said plurality of threads to said amount of said system processor time utilized by said identified thread set as each of said plurality of threads expire, wherein each of said plurality of threads is a path of execution such that multiple of said plurality of threads can be executed simultaneously; and~~
- ~~— storing said system processor time for said identified thread set in said hash table, thereby tracking said system processor time of said target agent.~~

21. (currently amended) The method of claim ~~120~~ further comprising:  
computing statistics for said plurality of threads~~identified thread set~~.
22. (currently amended) The method of claim ~~210~~ further comprising:  
rank ordering said plurality of threads.
23. (currently amended) The method of claim 22 further comprising:  
providing said plurality of threads ~~identified thread set~~ to a display device.

Please add the following new claims:

24. (new) A computer system including a computer readable storage medium having program code stored thereon for monitoring system processor resources utilized by a software agent operating in said computer system, wherein said agent comprises an executable sequence of instructions, said program code comprising:

program code for identifying said agent, wherein said identifying is performed by a CPU resource tracking process determining that said agent is running;

program code for associating, by said CPU resource tracking process responsive to said identifying of said agent, an agent identifier with said agent, wherein said agent is one of a plurality of software agents operating in said computer system, and wherein said agent identifier uniquely identifies said agent within said plurality of software agents operating in said computer system;

program code for initiating, responsive to said identifying of said agent, an agent lifetime timer for measuring an operating interval of said agent during which said agent is running in said computer system;

program code for determining said operating interval using said agent lifetime timer by identifying a start time at which said CPU resource tracking process determined that said agent is running and a completion time at which said CPU resource tracking process determines that said agent has expired, and computing said operating interval as the difference between said starting time and said completion time;

program code for calculating an amount of said system processor resources utilized by said agent during said operating interval at least in part by detecting a plurality of threads created by said agent during said operating interval, wherein said calculating said amount of said system processor resources utilized by said agent during said operational interval further includes calculating CPU usage for each of said plurality of threads and adding said calculated CPU usage for all of said plurality of threads together to determine at least a portion of said amount of system processor resources utilized by said agent during said operating interval, wherein said calculating said CPU usage for each of said plurality of threads is performed responsive to each respective one of said plurality of threads expiring, and wherein said adding said calculated CPU usage for all of said plurality of threads together is performed by adding said calculated CPU utilization for each respective one of said plurality of threads responsive to each respective one of said plurality of threads expiring, wherein each of said plurality of threads is a path of execution such that multiple of said plurality of threads can be executed simultaneously; and

program code for storing said operating interval, said amount of system processor resources utilized by said agent during said operating interval and said agent identifier in a computer-readable memory.

25. (new) The system of claim 24, wherein said computer-readable memory includes a hash table.

26. (new) The system of claim 24, said program code further comprising:

program code for associating said operating interval and said agent identifier with other operating intervals and agent identifiers associated with other ones of said plurality of software agents operating in said computer system.

27. (new) The system of claim 26, said program code further comprising:

program code for filtering said plurality of software agents according to predefined filtering criteria to produce a filtered set.

28. (new) The system of claim 27, said program code further comprising:  
program code for rank ordering said filtered set.
29. (new) The system of claim 28, said program code further comprising:  
program code for making said filtered set available to a display device.
30. (new) The system of claim 24, said program code further comprising:  
program code for determining a corrective measure for at least one member of said  
filtered set having an amount of system processor resources utilized that exceeds a predetermined  
threshold.
31. (new) The system of claim 30, said program code further comprising:  
program code for displaying said corrective measure on a display device.
32. (new) The system of claim 30, wherein said corrective measure is implemented by said  
system.
33. (new) The system of claim 24, said program code further comprising:  
program code for running a collection probe to determine if a total amount of consumed  
system processor resources exceeds a predetermined threshold; and  
program code for performing said initiating step when said total amount of consumed  
system processor resources exceeds said configuration threshold.
34. (new) A computer program product, comprising:  
a computer readable storage medium having program code stored thereon for monitoring  
system processor resources utilized by a software agent operating in said computer system,  
wherein said agent comprises an executable sequence of instructions, said program code  
comprising:  
program code for identifying said agent, wherein said identifying is performed by  
a CPU resource tracking process determining that said agent is running;

program code for associating, by said CPU resource tracking process responsive to said identifying of said agent, an agent identifier with said agent, wherein said agent is one of a plurality of software agents operating in said computer system, and wherein said agent identifier uniquely identifies said agent within said plurality of software agents operating in said computer system;

program code for initiating, responsive to said identifying of said agent, an agent lifetime timer for measuring an operating interval of said agent during which said agent is running in said computer system;

program code for determining said operating interval using said agent lifetime timer by identifying a start time at which said CPU resource tracking process determined that said agent is running and a completion time at which said CPU resource tracking process determines that said agent has expired, and computing said operating interval as the difference between said starting time and said completion time;

program code for calculating an amount of said system processor resources utilized by said agent during said operating interval at least in part by detecting a plurality of threads created by said agent during said operating interval, wherein said calculating said amount of said system processor resources utilized by said agent during said operational interval further includes calculating CPU usage for each of said plurality of threads and adding said calculated CPU usage for all of said plurality of threads together to determine at least a portion of said amount of system processor resources utilized by said agent during said operating interval, wherein said calculating said CPU usage for each of said plurality of threads is performed responsive to each respective one of said plurality of threads expiring, and wherein said adding said calculated CPU usage for all of said plurality of threads together is performed by adding said calculated CPU utilization for each respective one of said plurality of threads responsive to each respective one of said plurality of threads expiring, wherein each of said plurality of threads is a path of execution such that multiple of said plurality of threads can be executed simultaneously; and



program code for storing said operating interval, said amount of system processor resources utilized by said agent during said operating interval and said agent identifier in a computer-readable memory.